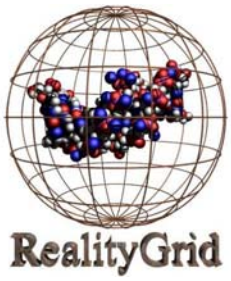
RealityGrid Steering Application User Manual



Mark Riding and Andrew Porter.

SVE Group, Manchester Computing, University of Manchester.

29th April 2004.

Change Log

Author	Comments	Date
Andrew Porter	As released with v.1.1 of the RealityGrid steering software.	29th April 2004

Acknowledgements

The RealityGrid Steerer is mainly the work of Sue Ramsden, with some further enhancements by Mark Riding and Andrew Porter. The steerer is built using the RealityGrid Steering API, by Andrew Porter.

Introduction

The RealityGrid Steerer is a generic user interface for performing computational steering on (computational) jobs which have been built using the RealityGrid Steering API. This document aims to give new users an overview of how to use the steerer, and to explain the functionality which it offers.

Starting the Steerer

Environment set-up

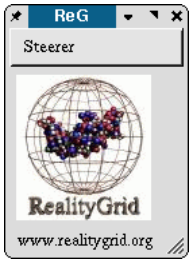
The steerer requires that some environment variables be set before it is executed. The main one of these is REG_STEER_HOME which should contain the path of the directory containing the steering library (*e.g.* /home/zzcguap/projects/reg_steer_lib). If the steering library is not available then any valid path will suffice.

If local (file-based) steering is being used then REG_STEER_DIRECTORY should contain the path to the directory to use for steering communication. This should match the value of REG_STEER_DIRECTORY passed to the steered application.

Alternatively, if remote steering (using Grid Services) is to be used then REG_SGS_ADDRESS should be set to the address of the Steering Grid Service of the steered application. The value given in this environment variable can be edited from within the steerer (see later) or, alternatively, specified on the command line.

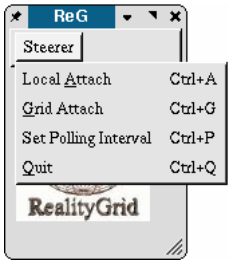
Launching

Run the steering application by executing the command 'steerer' at the command prompt (assuming that it is in your path). You will be presented with a simple start-up screen:



Steerer startup screen

The Steerer menu



Steerer menu

The Steerer menu has four options:

- Local Attach — attach to a local steerable job, communicated with by means of a local, shared file system. Shortcut Ctrl+A
- Grid Attach — attach to a steerable job located on 'any' machine, communicated with by means of a Grid-Service interface. Shortcut Ctrl+G

- Set Polling Interval — Set how often (in seconds) the steerer polls the computational job. Shortcut Ctrl+P
- Quit — Quits the application. Shortcut Ctrl+Q

The Polling Interval is an important parameter since it controls how often the steerer checks for status messages from an attached application. If the steerer fails to "keep up" with the application then status messages will be lost which may result in strange behaviour. For applications that generate many status reports per second, the value of the "STEERING INTERVAL" steerable parameter should be increased in order to reduce the frequency with which the application generates status reports. (This steerable parameter is automatically generated by the steering library and is thus available for all applications that use the RealityGrid steering library.)

Attaching to a steerable job

Selecting Local Attach from the Steerer menu will take you straight to the main RealityGrid Steerer window. Should the steerer fail to find an application to attach to then the Steerer startup screen will be displayed with a "Failed to attach" message displayed. If this occurs then check that the REG_STEER_DIRECTORY environment variable in the steerer's environment matches that set for the simulation and that the application is actually running.

Selecting Grid Attach will bring up a small dialog box first:

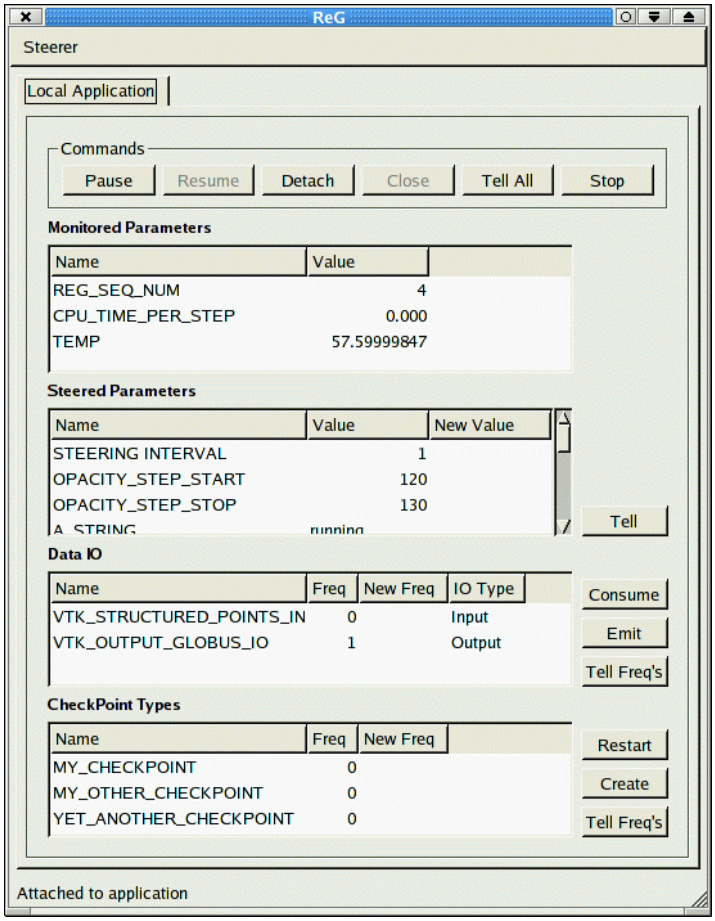
Grid Attach dialog

The dialog lists available RealityGrid applications and their addresses (handles) in a table. Entries in the Handle column are fully editable. By selecting an entry in the table and clicking on the Attach button, a Steerer is created for the chosen application. Should the attach fail then check that the handle you've specified is correct and that the application you are attempting to connect to is up and running.

It is also possible (on Linux systems) to specify the handle to connect to on the command line when launching the steerer.

Using the Steerer

The Main Window



Steerer main window in local-steering mode

The Main Window is split into five conceptual sections:

- Control Buttons
- Monitored Parameters
- Steered Parameters
- Data IO
- CheckPoint Types

Control Buttons

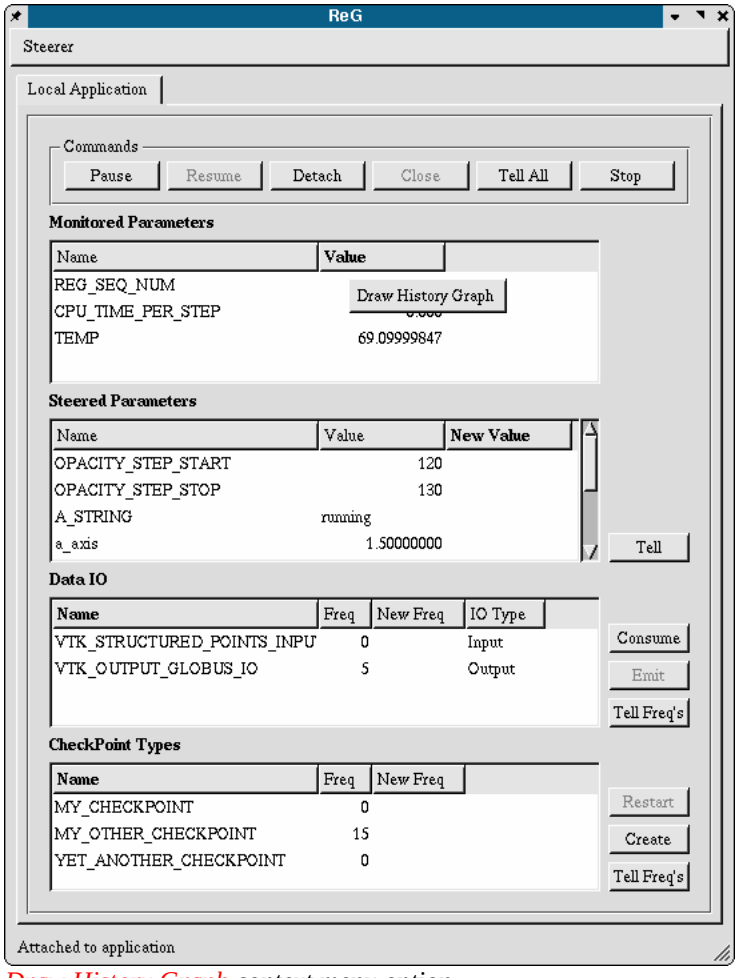
The Control Buttons panel manages overall control of the steerable job's state. Different buttons will be greyed out and unavailable depending on a job's state; for instance, jobs that are not running cannot be stopped.

- Pause button — pauses a running job
- Resume button — causes a paused job to resume
- Stop button — ends the job
- Detach button — detaches the steerer from the job, leaving the job running
- Close button — closes the main window of the steerer, returning to the startup screen
- Tell All button — causes the steerer to pass any user updated parameters to the job, see Steered Parameters section, below

Monitored Parameters

The Monitored Parameters table lists parameters which are exposed by the steerable job, but are not themselves directly steerable. An example of a monitored parameter for a simulation job might be time per simulation step. Such a parameter is changed only by the steered application itself, not the user.

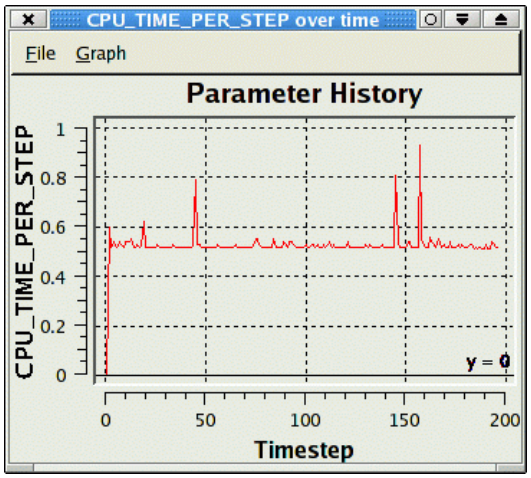
The Steerer allows graphs to be drawn of monitored parameters. By right clicking on a monitored parameter, a context menu appears allowing the user to select a 'Draw History Graph' option.



Draw History Graph context menu option

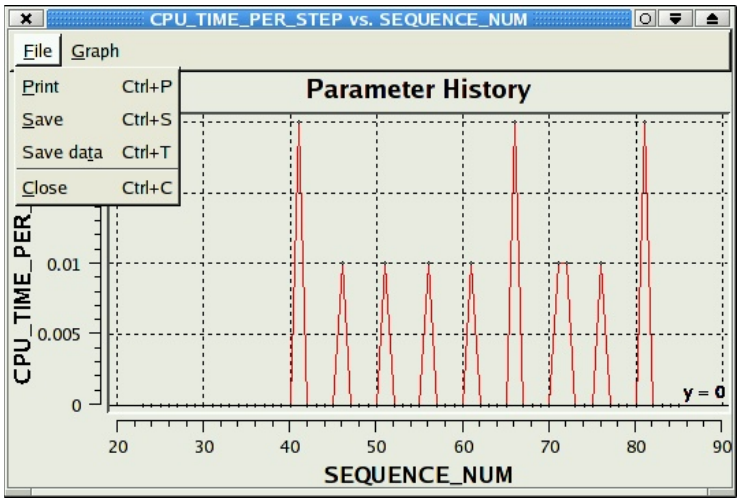
***Parameter History Graph*s**

Selecting the Draw History Graph option from the context menu causes a new window to appear, containing a simple 2D line plot of the selected parameter's history. The graph is constructed simply by connecting adjacent ordinates with straight lines – no fitting is performed.



Example Parameter History Graph

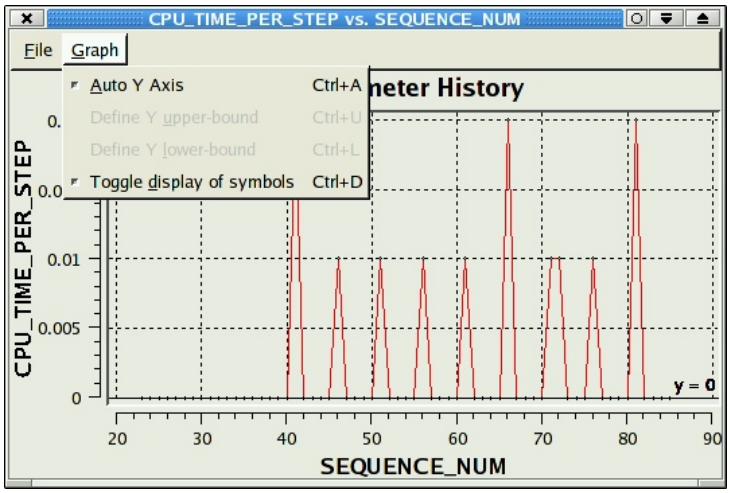
There are two menu's available in this window, the File menu, and the Graph menu.



Parameter History Graph: File menu

The File menu has four options:

- Print requests a print-out of the image portion of the window. Shortcut Ctrl+P
- Save saves the image portion of the window to disk. Shortcut Ctrl+S
- Save data saves the raw data to file (pops up a file browser). Shortcut Ctrl+t
- Close closes the window. Shortcut Ctrl+C

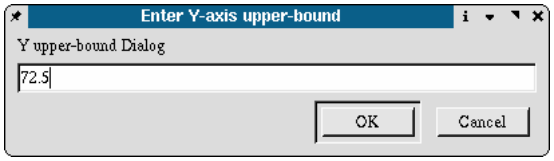


Parameter History Graph: Graph menu

The Graph menu also has four options:

- Auto Y Axis — allows the application to automatically determine the extents of the y-axis. This menu item toggles on and off. Shortcut Ctrl+A
- Define Y upper-bound — if Auto Y axis is not selected, this menu item becomes available. By selecting it, a dialog appears (see below) allowing you to enter an upper bound for the Y axis. If Auto Y Axis is reselected, then this value is ignored. Shortcut Ctrl+U
- Define Y lower-bound — as 'Define Y upper-bound', but for the lower bound. Shortcut Ctrl+L
- Toggle display of symbols — enables/disables the display of symbols at data points. Shortcut Ctrl-D

Symbols are displayed by default if the graph is of an appropriate scale – if resolution does not permit then they are automatically hidden.



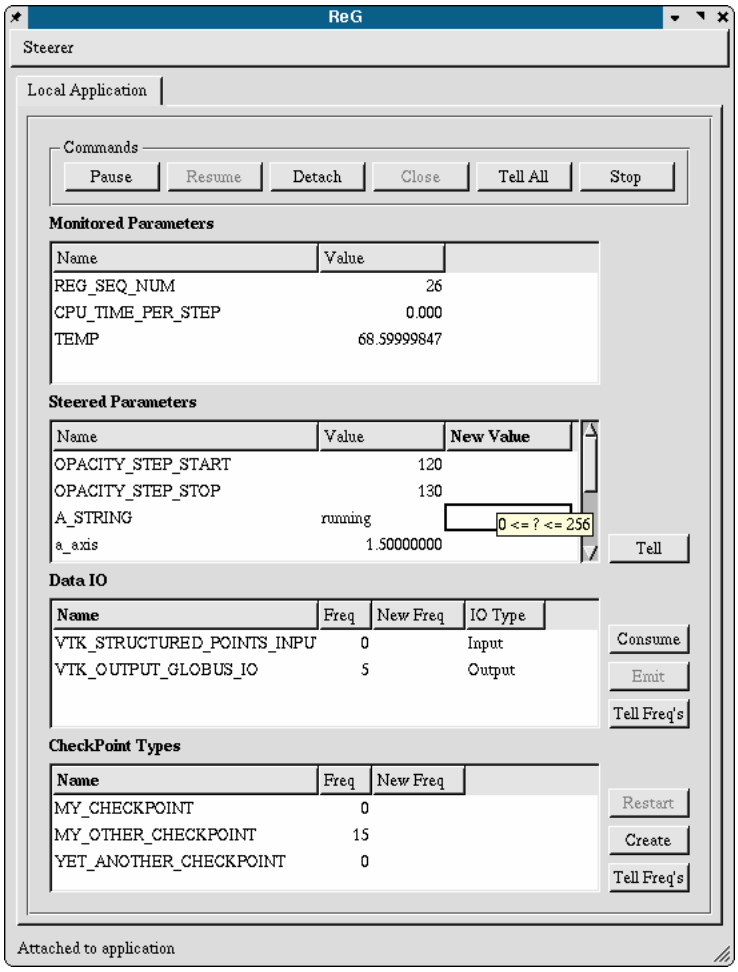
Y axis upper-bound dialog

Note that the steering client automatically logs the values of all monitored parameters (irrespective of whether they are being plotted) and therefore if it is connected to a running simulation for an extended period it may accumulate a considerable amount of data. Displaying a history plot of a large data set obviously uses some computational resource, particularly if the plot is being updated frequently. Therefore it is recommended (in order to maximise the responsiveness of the steering client) that history plots be closed while not in use if the steering client is being used to continuously monitor long runs.

Steered Parameters

The Steered Parameters list box contains a list of all steerable parameters exposed by the current job. The first column, 'Name,' lists the names of each of the parameters, the second, 'Value,' shows the current value for each parameter, and the third, 'New Value,' allows you to enter new parameters. Hovering the mouse over an entry in the third column brings up a tool tip showing the minimum and maximum allowable values for that parameter. If the steerable parameter is a string then the maximum allowable value gives the maximum length of the string.

New values are only propagated to the running job once the entry has been completed (by pressing Enter or leaving the cell via the cursor keys) and the 'Tell' button is clicked. When this happens the entries in the 'Value' column are updated to reflect the new values, and any entries in the 'New Value' column are cleared. This action is also performed if you click on the 'Tell All' button in the Commands button panel. Note that the entries in the 'Value' column are only updated upon confirmation from the application – this thus provides confirmation that it has received the instruction(s).

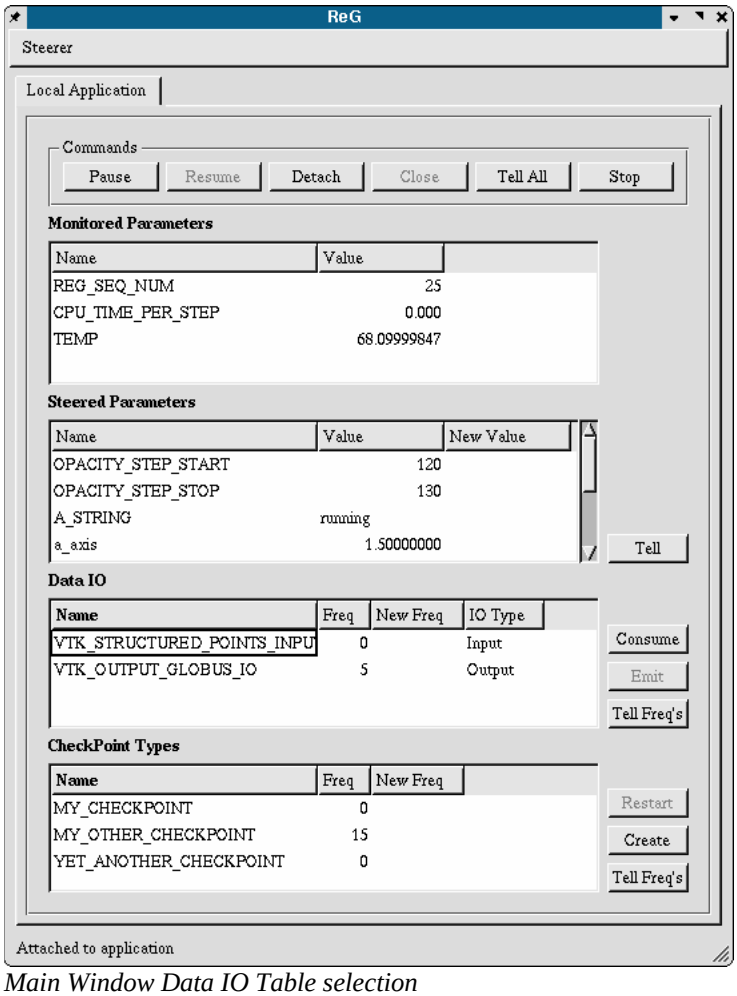


Steered Parameters tool tip

Data IO

The Data IO table lists all data input and output channels supported by the current job. The 'Name' column lists the name of each channel, the 'Freq' column lists how often data is written to or read from the channel (in time steps), the 'New Freq' column allows you to specify a new frequency for each item, and the 'IO Type' column indicates whether the channel is for input or output. Each individual entry is either an input or an output, not both. As with the steered parameters table, updates made are not propagated to the current job until either the 'Tell Freq's' button or the 'Tell All' button in the Commands button panel is clicked.

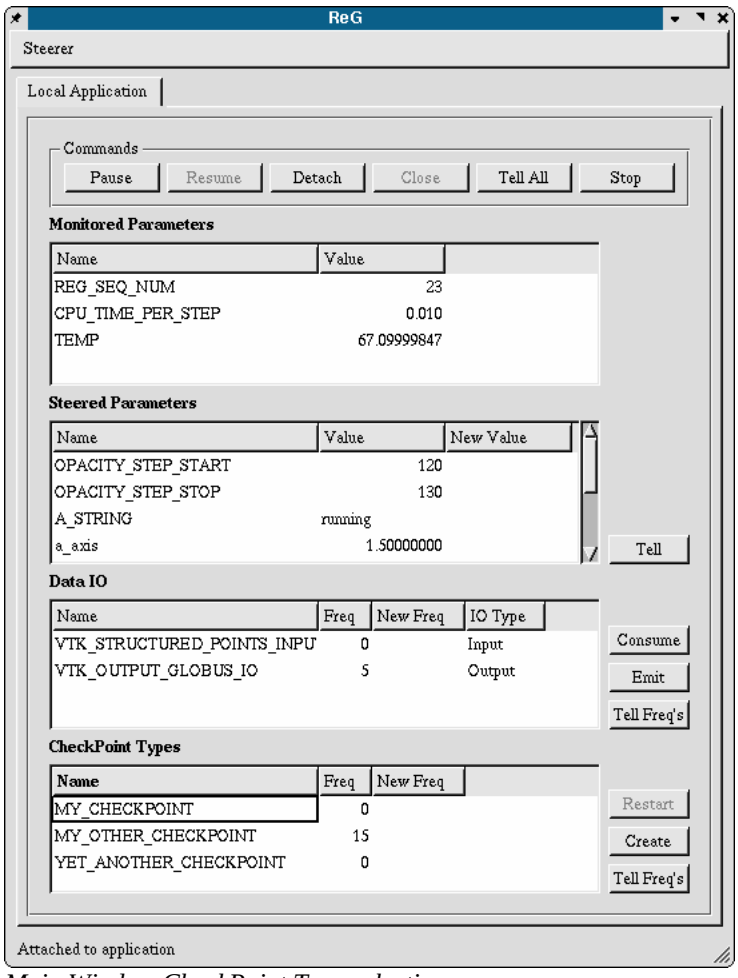
It is possible to manually request that data be emitted/consumed, in addition to the automatic events. To do so, select an entry from the table, and click on the 'Consume' or 'Emit' button at the right of the table. Only one of the buttons will be available for each entry, depending on whether the data channel is for input or output.



Main Window Data IO Table selection

CheckPoint Types

The CheckPoint Types table lists the types of checkpoint that the running job supports. There can be many instances (checkpoints) of each checkpoint type, and a job may expose several checkpoint types which differ in the parameters and data values they record. The majority of jobs will only need a single checkpoint type.



Main Window CheckPoint Type selection

The 'Name' column lists the names of each checkpoint type, the 'Freq' column lists the frequency with which checkpoints are automatically taken (in time steps), and the editable 'New Freq' column allows you to enter new values for a checkpoint's frequency. As with the Steered Parameter values, new frequency values are only propagated to the running job when either the 'Tell Freq's' button or the 'Tell All' button in the Commands button panel is pressed.

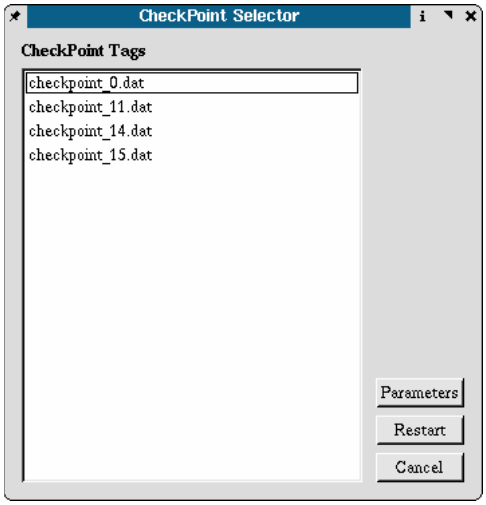
There are slight differences in the steerer main window depending on whether local (file-based) or remote steering (using the grid-service framework) is being performed.

Checkpointing with local steering

Checkpoints (checkpoint type instances) can be created or used to perform a restart by means of the 'Create' and 'Restart' buttons to the right of the main checkpoint table. These buttons will be

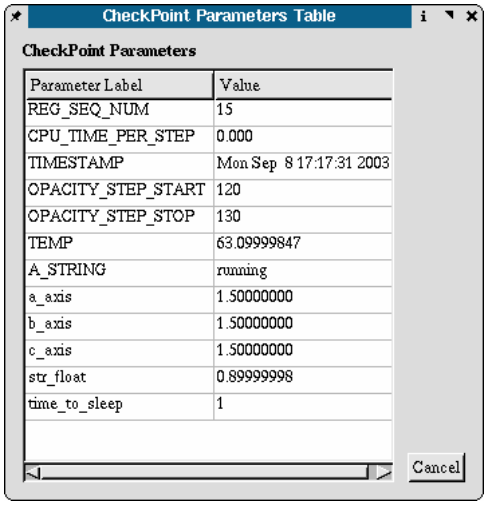
greyed out and unavailable by default, only becoming active if an appropriate checkpoint type has been selected.

After clicking on the Create button for a valid checkpoint type, the application will be instructed to create a checkpoint instance of the selected checkpoint type.
After clicking on the Restart button for a valid checkpoint type, a dialog is presented listing all the checkpoint instances for that type. You can select a checkpoint instance from the list, and instruct the application to restart from it by clicking on the 'Restart' button. Clicking on 'Cancel' quits the dialog without doing anything.



CheckPoint Selector dialog

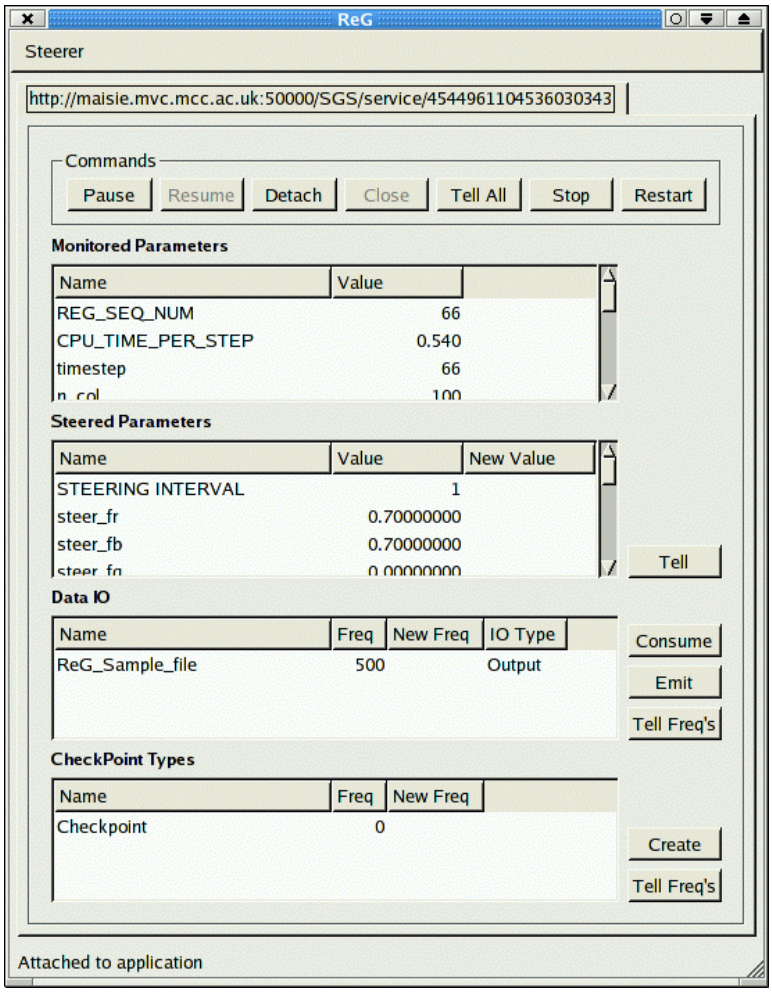
In order to help differentiate between checkpoint instances, it is possible to inspect the parameters of each by clicking on the 'Parameters' button. This brings up another dialog, which lists the checkpoint instance's parameters and their values at the time the checkpoint was taken. It is possible to have several Checkpoint Parameters Table dialog's open at the same time, so that different checkpoint instances can be compared and contrasted.
Again, clicking on the 'Cancel' button closes the dialog.



CheckPoint Parameters Table dialog

Checkpointing with remote steering

When remote steering is in operation, the main window of the steerer and the way in which a restart is performed is slightly different.



Steerer main window in remote-steering mode

A checkpoint is still created by selecting the appropriate checkpoint type from the checkpoint table and clicking the `Create' button. However, the `Restart' button is now located in the panel at the top of the window – this is because it is no longer necessary to select a checkpoint type before pressing it. Instead, on pressing `Restart,' the user is prompted to enter the Grid Service Handle of a suitable checkpoint. How the user obtains this GSH is beyond the scope of this document since it relies upon the user browsing a checkpoint tree (using either a web interface or some tool such as the RealityGrid launching wizard).

Index

A

Attach · 2

C

CheckPoint Types · 10
Checkpointing with local steering · 10
Checkpointing with remote steering · 12
Control Buttons · 5

D

Data IO · 9

E

Environment set-up · 2

I

Introduction · 1

M

Monitored Parameters · 5

P

Parameter History Graph · 6
Polling Interval · 3
Printing graphs · 6

S

Starting the Steerer · 2
Steered Parameters · 8